

RDD-100 and the Systems Engineering Process

Robert D. Averill, Systems Engineering Office, AMSD, IOG

Efforts to implement an effective systems approach to NASA programs are in progress Agency-wide ¹. At Langley we are trying to define an enhanced systems engineering process for in-house flight projects to assure that each system will achieve its goals with quality performance and within planned budgets and schedules. An effective systems engineering approach applied throughout the project life cycle can help Langley produce a better product. This paper will show how this can be done by utilizing a systems engineering process in combination with available software tools such as RDD-100 ². To accomplish this, I will, first, briefly discuss the systems engineering process and then show how RDD-100 has been applied as a pilot effort in the early phases of the SABER ³ instrument development.

(Chart 2) The objective is to show you how RDD-100 can be used as a systems engineering tool throughout the project life cycle and to challenge you to consider using this tool with your project team.

(Chart 3) Systems engineering may mean many different things to different people but this is the way it is defined in the Langley Systems Engineering Handbook ⁴ which is currently pending publication. The systems engineering process is really the key to how we approach the problem. There are many different procedures, methodologies, and models being used for systems engineering. It is important that each project define how systems engineering will be managed and conducted throughout the project life cycle.

(Chart 4) The Systems Analysis and Design Procedure is proposed for use at Langley during the Formulation Phases of the project when the systems engineering activity is the most intensive. This procedure provides a focused and structured systems engineering method and is a problem solving approach which can be tailored to project needs.

(Chart 5) The Systems Analysis and Design Procedure is a ten step process applied iteratively during each phase of the project. The concentric circles represent each phase; for example, the inner circle symbolizes the Pre-Phase A effort which has the purpose of quickly assessing the feasibility of a proposed project to determine if it justifies further development. The detailed activities of each step of the process are developed in more detail in LHB 7122.1. However, they can be quickly summarized as follows. The Initialization step includes a management decision to initiate the study and provide skills and resources necessary to do the job on a timely basis. The determination of User Needs and Goals is perhaps the most important step in scoping the effort; this leads directly to a definition of Systems Requirements to achieve the goals. Performance Measures are defined to provide a quantitative standard to assess system performance. Next, potential System Concepts are generated, Analyzed, and Ranked to determine system feasibility. Further Systems Development may be needed to bring the proposed system approaches to the level of maturity desired for this initial stage of development. The final step in the process provides for technical and management reviews to assess the status of the development. This represents a Decision Point which will determine if

the system will repeat the iterative development process or pass to the next phase of project development. It is believed that the use of such a customized systems engineering process with well defined tasks, products, and controls will help the Project Team perform most effectively. It should be emphasized that the systems engineering process is a team effort and is dependent upon project teamwork and communication throughout the process.

(Chart 6) The goal of the process is to enhance communication between different technical disciplines on the project. For example, the relationship between systems engineering and software engineering is vital to the success of the project. These two groups must work closely together to define their mutual information needs. Are the typical systems engineering "products" in the left column useful to the software engineering function? Are the typical software engineering "products" in the right column a logical and related extension of the systems engineering requirements? The project can operate most efficiently if a common technical language is used by all of the project team.

(Chart 7) There are currently available several computer aided systems engineering tools which propose to provide a common technical language for use throughout the project life cycle. One of these is the Object Modeling Technique developed by General Electric ⁵ and currently being marketed as StP/OMT ⁶. This tool is being evaluated for use at Langley but is not currently implemented. The RDD-100 tool is being used in the Systems Engineering Office at Langley. RDD-100 utilizes an object oriented methodology with a symbolic language designed to be useful to all technical disciplines.

(Chart 8) RDD-100 is an extension of the earlier Entity-Relationship Model (developed originally for information modeling use ⁷) into an object modeling concept. The power of the RDD-100 concept is that the Elements (Entities) are linked by binary relationships such that changes to any element are transferred to its related Elements; thus continuously updating the database. The tool also provides for requirements tracking throughout the system life cycle. Another powerful feature of RDD-100 is its modeling capability.

(Chart 9) The Integrated System Model is an evolutionary development which begins with the most rudimentary concept of system objects and progressively evolves into a complete model representing overall system dynamic performance. This provides continuity through the project life cycle and offers a "seamless" transition from phase-to-phase.

(Chart 10) We will now present a brief overview of RDD-100 capabilities.

(Chart 11) RDD-100 is a menu driven application and provides ready access to all of its features. It can be seen that the emphasis of the program is on system Elements. The Multi-Element View and the various Editors permit easy manipulation and editing of the system Elements.

(Chart 12) An example of the Multi-Element View concept is the SABER Requirements hierarchy. The Element-Relationship aspect is shown as, for example, Operational Objective: Interface Constraints **incorporates** Operational Objective: Instrument Mass.

(Chart 13) Shown here is a section of the requirements Custom Hierarchy which provides a visual display of the relationships between requirements.

(Chart 14) The modeling capability of RDD-100 is implemented by Behavioral Diagrams which incorporate all of the system functional and dynamic relationships on one diagram. This is a major advantage over other concepts which separate, for example, control and data functions on two unsynchronized models. The RDD-100 approach provides one self-contained Integrated System Model which demonstrates system dynamic response.

(Chart 15) This is the overall SABER Operational Model based on the five key objects selected for the system: User, Ground Station, Spacecraft, SABER Instrument, and Atmospheric Scene. The purpose of this Operational Model is to demonstrate the flow of top level control and data messages.

(Chart 16) The SABER instrument is shown here in more detail with the operational functions of current interest. The behavioral diagram includes Time Functions, Time Items, and, in this case, an Iterate Function, represented by the loop, which repeats the scan sequence for a specified number of cycles.

(Chart 17) The scenario shown is a running model and can be evaluated by the Dynamic Verification Facility. The system runs on an arbitrary time base which can represent any desired time scale. Various functions can be selected to display an Events Transcript, Time Lines, and System Resources. The Facility identifies any dynamic inconsistencies in the model.

(Charts 18 & 19) Shown are sections of the Event Transcript showing the beginning and end of the run.

(Charts 20 and 21) Shown are the Function Time Line and a history of the Scene Radiance resource. The instrument, in this example, accumulates ten data samples and then transmits them to the spacecraft.

(Charts 22 & 23) The Summary concludes that the use of a structured systems engineering process in conjunction with a powerful computer aided systems engineering tool is believed to provide the most effective approach to achieving project success at LaRC.

- 1 NASA Systems Engineering Handbook, Draft, September 1992, JPL.
- 2 RDD-100 - Requirements Driven Development, Ascent Logic Corporation.
- 3 Sounding of the Atmosphere using Broad band Emission Radiometry.
- 4 LHB 7122.1, Systems Engineering Handbook for In-House Space Flight Projects.
- 5 Rumbaugh, James; et al: Object Oriented Modeling and Design. Prentice Hall, 1991.
- 6 StP/OMT - Software through Pictures/ Object Modeling Technique, Martin Marietta Advanced Concepts Center and Interactive Development Environments.
- 7 Sage, Andrew P., and Palmer, James D.: Software Systems Engineering. John Wiley and Sons, 1990.

Robert D. Averill

**RDD-100 and the
Systems Engineering Process**

The Role of Computers in LARC R&D

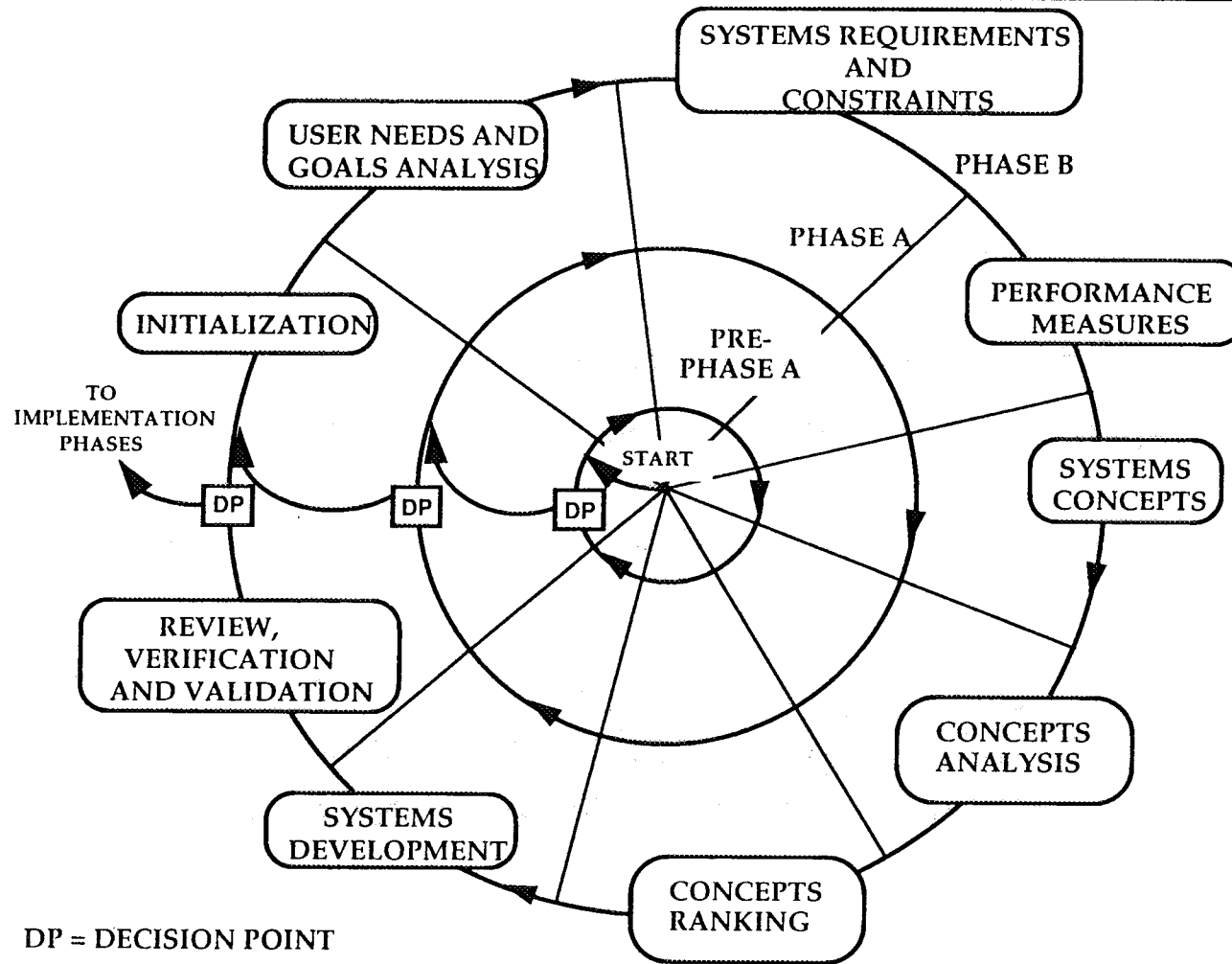
June 16, 1994

OBJECTIVE

- How RDD-100 can be used as a Systems Engineering Tool throughout the project life cycle.

- **Systems Engineering:** A function that guides the transformation of customer/user's needs into a flight system that meets technical performance requirements. . .
- The **objective of systems engineering** is to provide a robust system which satisfies the customer's technical performance objectives within the constraints of cost and schedule.
- The **systems engineering process** is the approach to achieve this objective and is defined in the Langley Systems Engineering Handbook (LHB 7122.1)

- **Systems Analysis and Design Procedure:** An interactive, analytical, step-by-step approach during the Formulation Phases of the project life cycle.
 - Pre-Phase A: Preliminary requirements and concepts analysis.
 - Phase A: Requirements definition and conceptual trades.
 - Phase B: Concept definition and preliminary design.



Bridging the Gap

(Need for Common Terminology)

Systems Engineering

- Goals Analysis
- System Requirements
- System Concepts
- Project Database
- Design Specifications
- System Model
- Performance Verification Matrix
- Other _____



Software Engineering

- Operations Model
- Requirements Model
- Control/Data Flow Diagrams
- System Architecture
- Methodologies
(SA/SD, JSD, OOA, ESP)
- V & V; Risk Analysis
- Data Dictionary; P-Specs
- Other _____

RDD - 100 Initiatives

- RDD - 100 is a product family developed by Ascent Logic to support the systems engineering process and improve program success.
- RDD - 100 objective is to enable concurrent engineering methodologies and support communication between engineering disciplines.
- RDD - 100 utilizes an object-oriented methodology with “bridges” to CASE tools such as Teamwork and Software through Pictures (StP)

RDD - 100 Systems Engineering Improvements

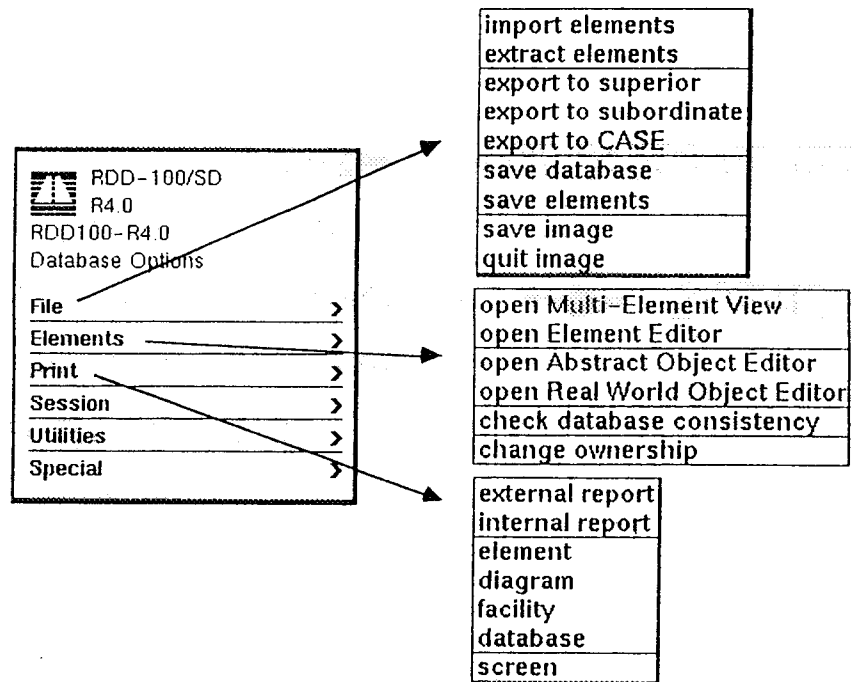
- Developing an executable notation for systems definition
 - Elements - class of objects
 - Relationships - links to other elements
 - Attributes - unique characteristics
- Maintain an interrelated database to record traceability and design decisions.
- Use of a systematic and automated documentation process.
- Requirements extraction, definition, allocation, and tracking.
- Increased use of modeling and prototyping to identify timing and interface issues.

Integrated System Model Provides An Evolutionary Modeling Representation Of The System

- Real World Object Model - system context and "objects".
- Data Model - external message flows.
- Operational scenarios - stimulus and response sequence of system elements.
- Conceptual Behavior Model - system model defining system objects and functions.
- Component Interconnection Model - defines system requirements allocations.
- Stimulus/Response Model - overall dynamic performance.

RDD - 100 Walk - Through

- Main Menu
- SABER Requirements.
- Hierarchy of Requirements.
- System Engineering Notebook → Report Writer.
- Operational Model → Dynamic Verification Facility.



The RDD - 100 System Designer Main Menu

Source: NASA2.txt

NUMBER.

DESCRIPTION. The overall goal of the SABER experiment is to improve understanding of the thermal, chemical, and dynamical structure and the energetics of the mesosphere and lower thermosphere by conducting global-scale measurements of kinetic temperature, radiatively and chemically significant minor species concentrations, and long-lived species for use as dynamical tracers.

[1] *documents* OperationalObjective: Infrared Remote Sensing

NUMBER. 1.0

DESCRIPTION. To apply the space-proven technique of infrared earth limb emission remote sensing to the virtually unexplored region above ~ 50 km.
Ref. Proposal par. 2.1, 4.0.

[2] *incorporates* OperationalObjective: Interface Constraints

NUMBER. 1.1

DESCRIPTION. To conform to the interface constraints of the TIMED spacecraft.

[3] *incorporates* OperationalObjective: Instrument Mass

NUMBER. 1.1.1

DESCRIPTION. To develop an instrument of <66 kg mass.

[3] *incorporates* OperationalObjective: Heat Rejection

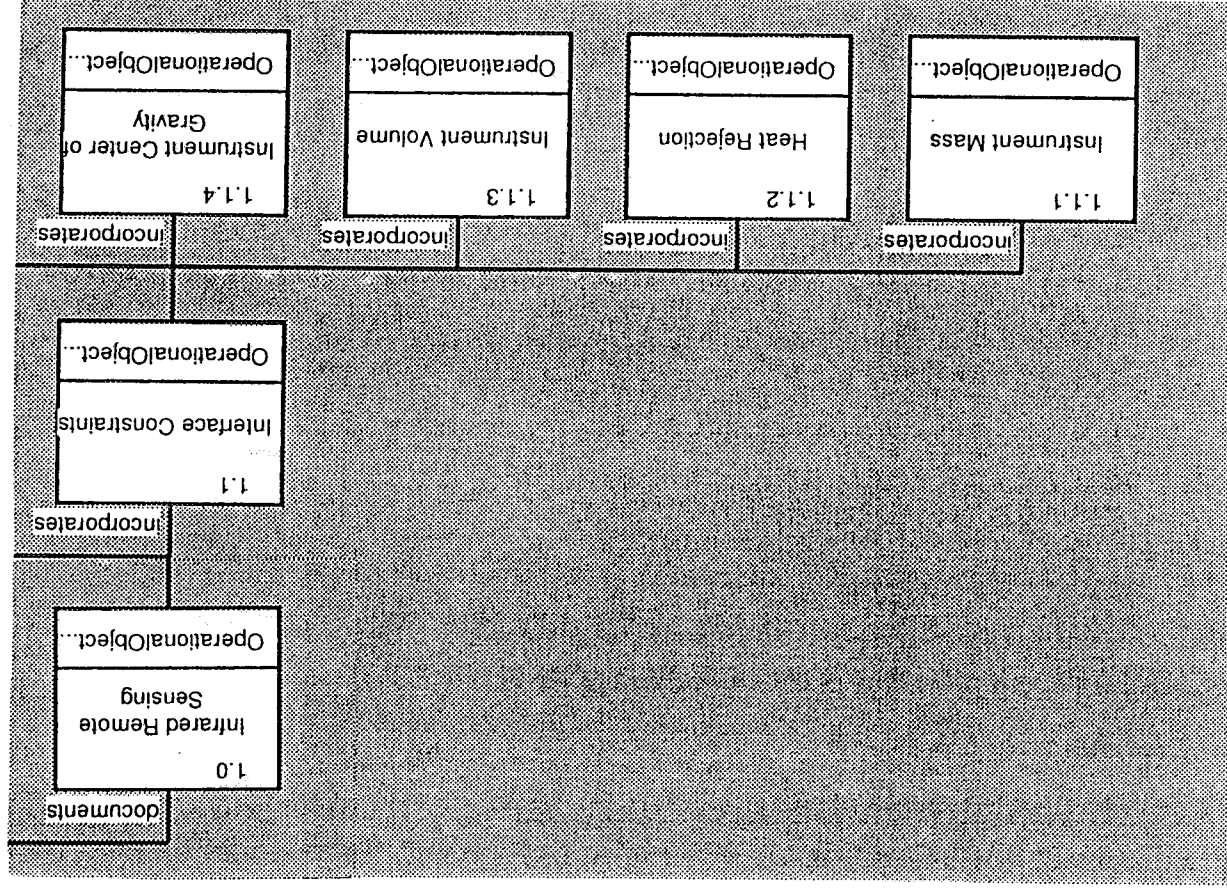
NUMBER. 1.1.2

DESCRIPTION. To develop an instrument requiring <50 watts of heat rejection to the spacecraft at a temperature of <290 K.

SABER Requirements

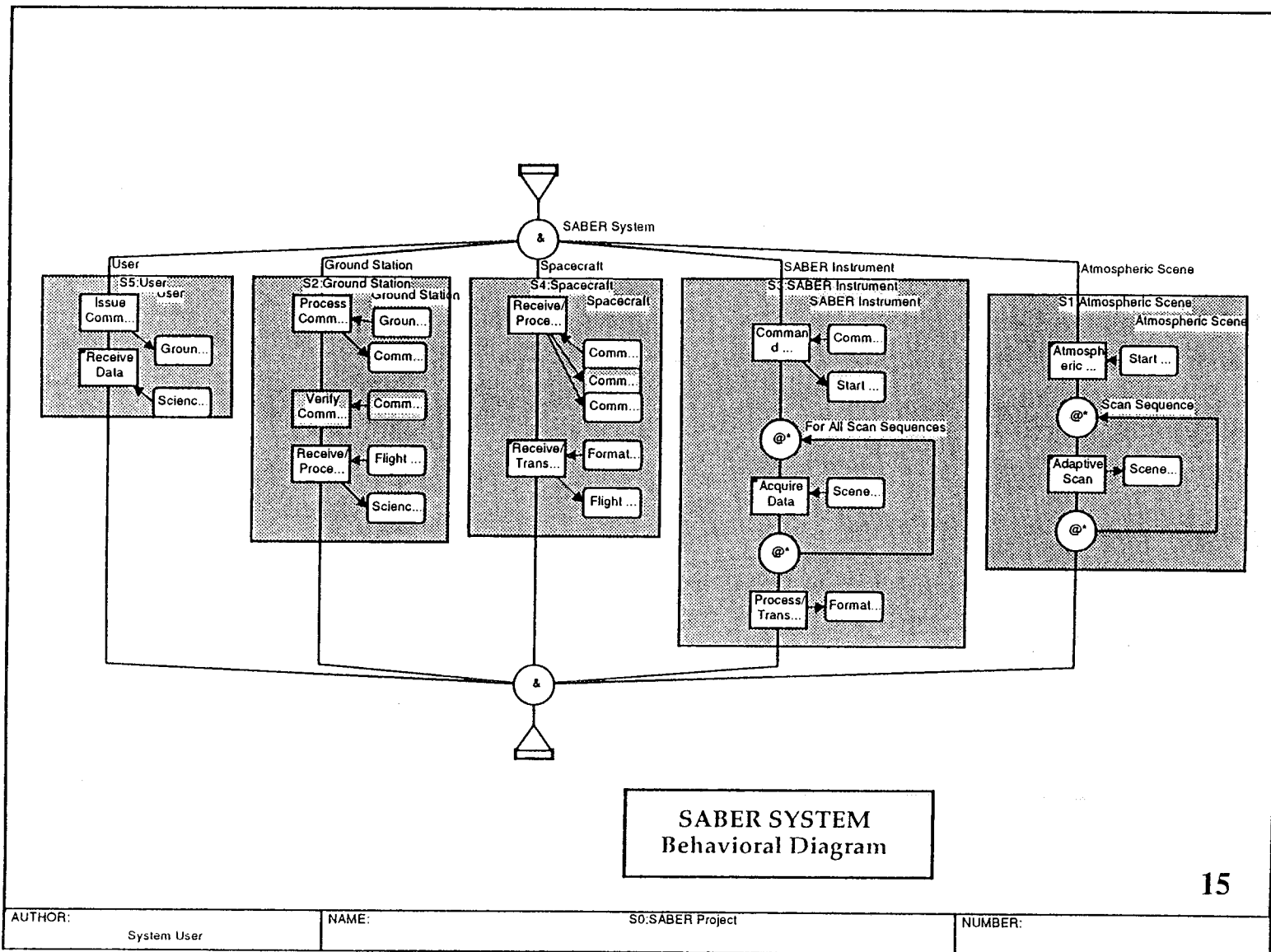
RDD - 100
Requirements Driven Development

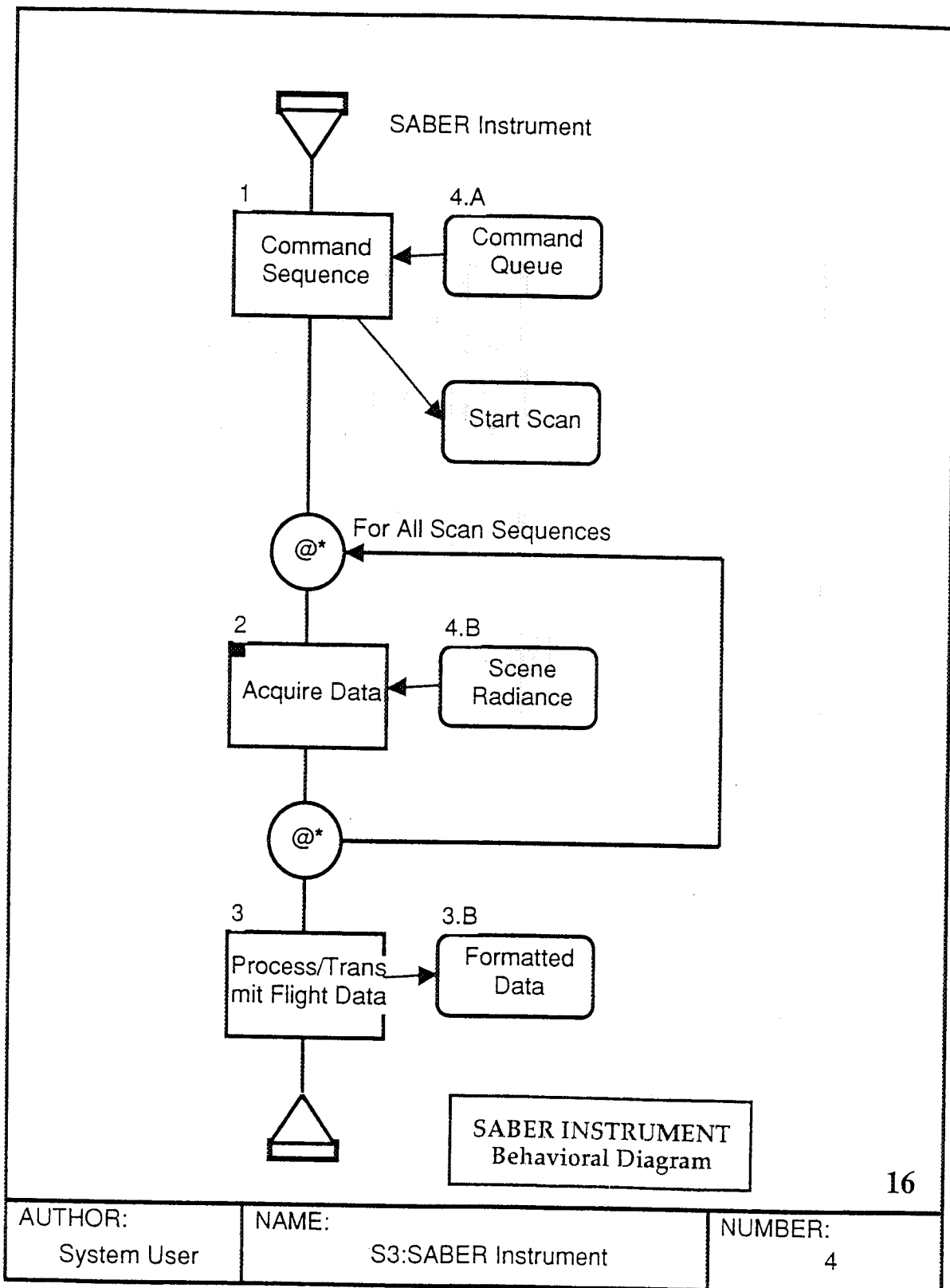
SABER Requirements Hierarchy



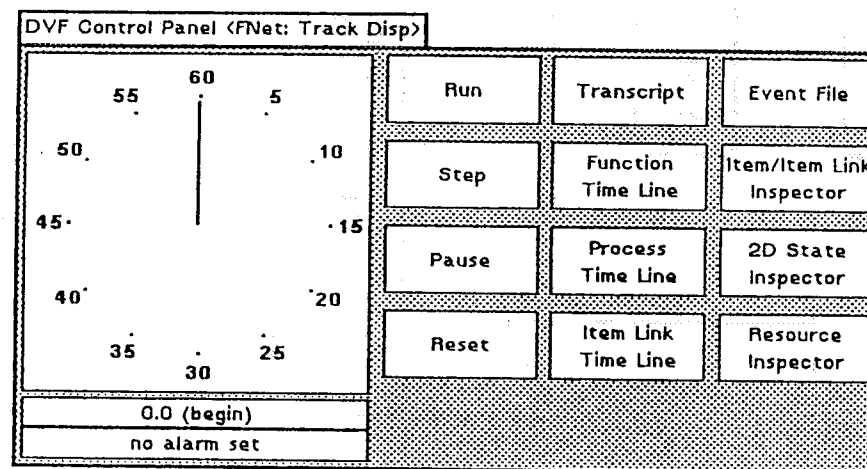
RDD - 100 Implemented by Behavioral Diagrams

- Defines system scenarios and interfaces.
- Verifies functionality and performance.
- Provides an Integrated System Model.
- Demonstrates system dynamic response.





DVF Control Panel



SABER Dynamic Verification Facility Control Panel

Event Transcript

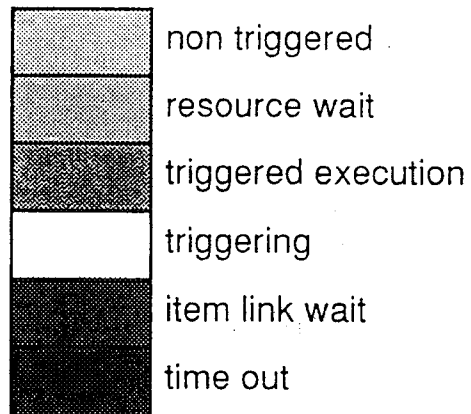
0.0 [S0:SABER Project] created
 0.0 [S0:SABER Project] creating process ('User')
 0.0 [S0:SABER Project] creating process ('Ground Station')
 0.0 [S0:SABER Project] creating process ('Spacecraft')
 0.0 [S0:SABER Project] creating process ('SABER Instrument')
 0.0 [S0:SABER Project] creating process ('Atmospheric Scene')
 0.0 [User] created
 0.0 [Ground Station] created
 0.0 [Spacecraft] created
 0.0 [SABER Instrument] created
 0.0 [Atmospheric Scene] created
 0.0 [User] (TimeFunction: Issue Commands) enabled
 0.0 [User] (TimeFunction: Issue Commands) proceeding
 0.0 [User] (TimeFunction: Issue Commands) working (10.0 10.0)
 0.0 [Ground Station] (TimeFunction: Process Command) enabled
 0.0 [Ground Station] (TimeFunction: Process Command) waiting for message
 0.0 [Spacecraft] (TimeFunction: Receive/Process Commands) enabled
 0.0 [Spacecraft] (TimeFunction: Receive/Process Commands) waiting for message
 0.0 [SABER Instrument] (TimeFunction: Command Sequence) enabled
 0.0 [SABER Instrument] (TimeFunction: Command Sequence) waiting for message
 0.0 [Atmospheric Scene] (TimeFunction: Atmospheric Radiance) enabled
 0.0 [Atmospheric Scene] (TimeFunction: Atmospheric Radiance) waiting for message
 10.0 [User] (TimeFunction: Issue Commands) sending message out to ('Timeltem' 'Ground Commands'
 nil nil 0 0 'Ground Station' 4067)
 10.0 [User] (TimeFunction: Issue Commands) ended
 10.0 [User] (TimeFunction: Receive Data) enabled
 10.0 [User] (TimeFunction: Receive Data) waiting for message
 10.0 [Ground Station] (TimeFunction: Process Command) received message ('Ground Commands' nil
 'User' 4067)

SABER Event Transcript (Start)

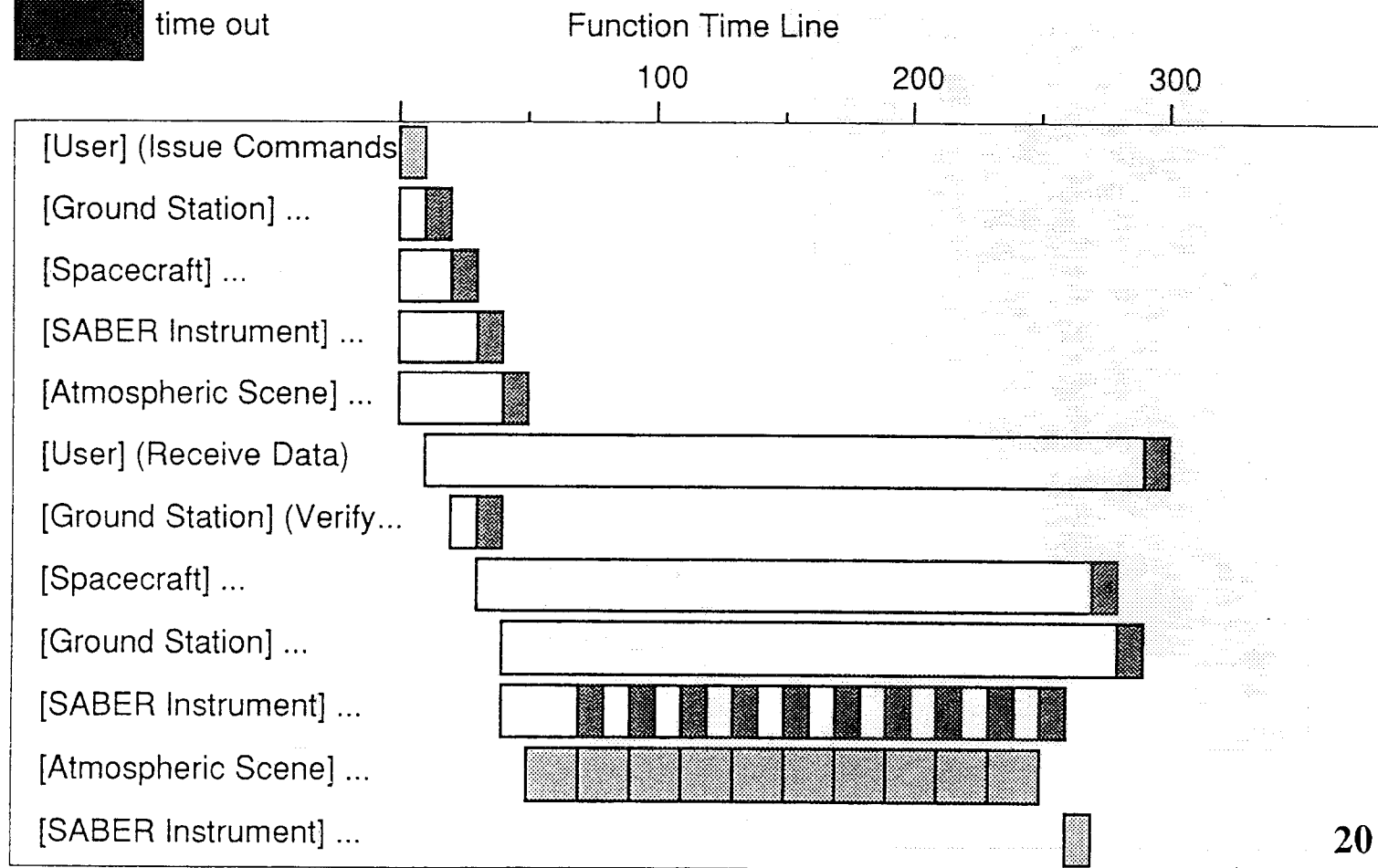
Event Transcript

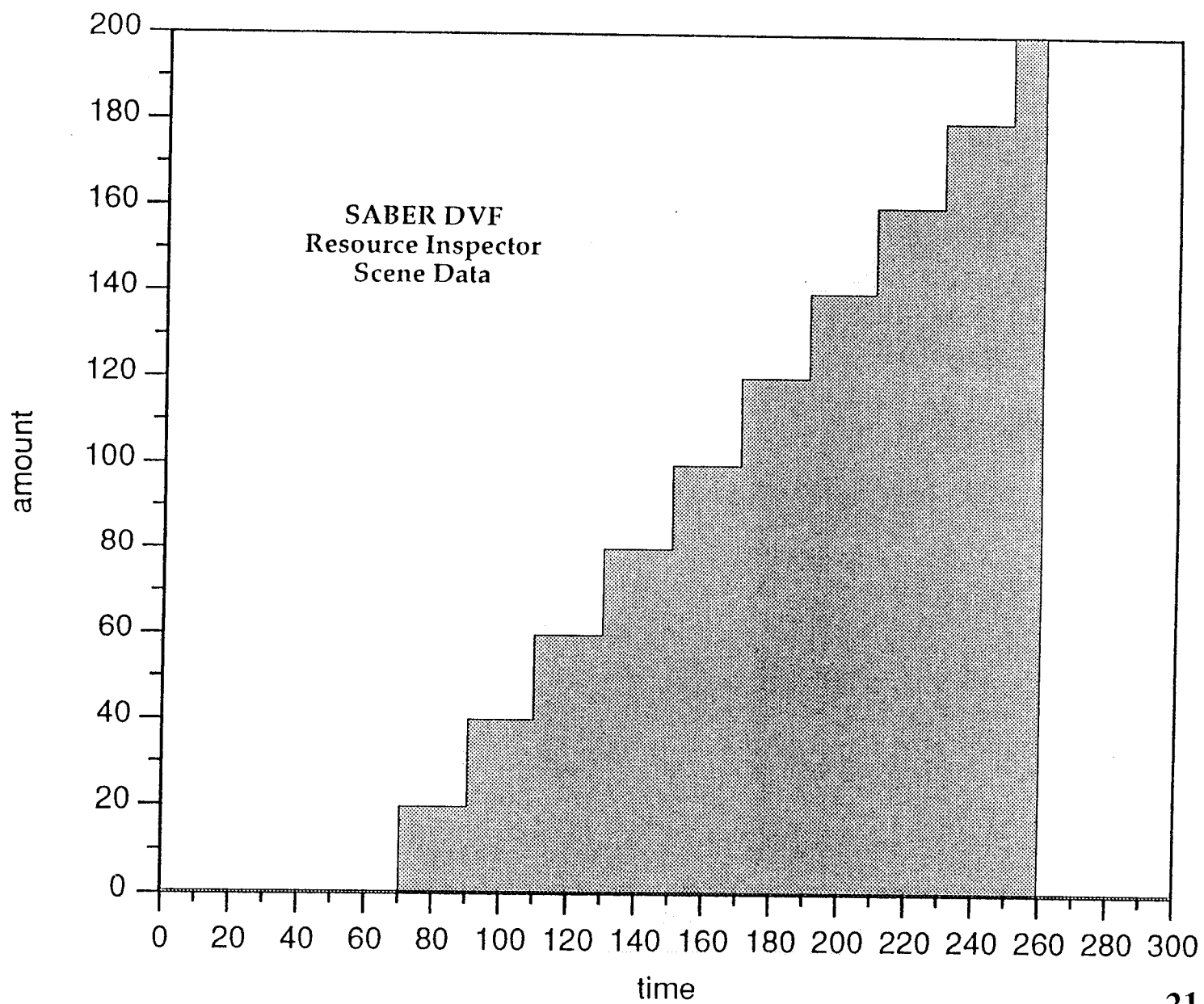
280.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) consuming resource ('Resource' 'Flight Data' 200.0 200.0)
 280.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) resources received ('Resource' 'Flight Data' 200.0 0.0)
 280.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) working (10.0 290.0)
 290.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) producing resource ('Resource' 'Science Data' 200.0 0.0)
 290.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) sending message out to ('Timeltem' 'Science Data' nil nil 0 0 'User' 980)
 290.0 [Ground Station] (TimeFunction: Receive/Process Flight Data) ended
 290.0 [Ground Station] terminated
 290.0 [User] (TimeFunction: Receive Data) received message ('Science Data' nil 'Ground Station' 980)
 290.0 [User] (TimeFunction: Receive Data) triggered
 290.0 [User] (TimeFunction: Receive Data) proceeding
 290.0 [User] (TimeFunction: Receive Data) consuming resource ('Resource' 'Science Data' 1.0 200.0)
 290.0 [User] (TimeFunction: Receive Data) resources received ('Resource' 'Science Data' 1.0 199.0)
 290.0 [User] (TimeFunction: Receive Data) working (10.0 300.0)
 300.0 [User] (TimeFunction: Receive Data) ended
 300.0 [User] terminated
 300.0 [S0:SABER Project] terminated

SABER Event Transcript (End)



SABER DVF Function Time Line





Summary

- The Langley Systems Engineering Process provides a well defined and effective approach to systems engineering.
- A Langley Systems Engineering Handbook (LHB 7122.1) will allow Project Managers and Systems Engineers to tailor the systems engineering process for their projects.
- The use of computer aided systems engineering tools will allow project teams to operate more productively.

Summary

RDD - 100 provides:

- A powerful tool for improving the systems engineering process.
- An interrelated database for the management of project requirements.
- A method for rapid simulation and modeling of system performance.
- A common technical language to improve communication between technical disciplines.
- A method for automated documentation and direct requirement linkage on projects.